

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-028182

(43)Date of publication of application : 04.02.1994

(51)Int.Cl.

G06F 9/38

(21)Application number : 04-096514

(71)Applicant : MATSUSHITA ELECTRIC IND CO LTD

(22)Date of filing : 16.04.1992

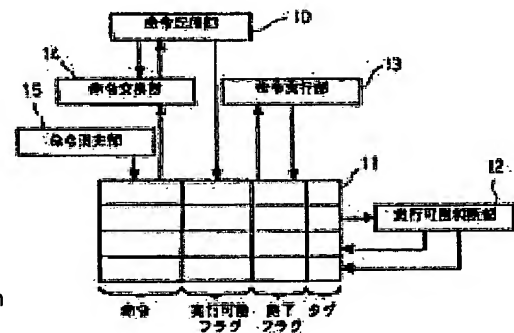
(72)Inventor : YAMAGUCHI RYUICHI

(54) INFORMATION PROCESSOR

(57)Abstract:

PURPOSE: To improve the instruction execution efficiency by lowering the frequency that an execution control part is full of unexecutable instructions and smoothly supplying instructions to the execution control part.

CONSTITUTION: An executable state judgement part 12 judges whether or not an instruction read in the execution part 11 can be executed and sets a tag of the execution control part 11 unless the instruction can be executed because of the relation of dependency on other instructions in the execution control part 11 which are being executed. When the instruction whose tag is set in the execution control part 11 can be replaced with another instruction in later execution order, an instruction replacement part 14 performs writing to an instruction storage part 10 so that the storage order between the both is changed. Consequently, when the same program is executed again, executable instructions are sent to the execution control part 11 prior to the unexecutable instructions.



(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-28182

(43)公開日 平成6年(1994)2月4日

(51)Int.Cl.⁵

G 0 6 F 9/38

識別記号

3 1 0 F 9193-5B

庁内整理番号

F I

技術表示箇所

審査請求 有 請求項の数 1 (全 5 頁)

(21)出願番号 特願平4-96514

(22)出願日 平成4年(1992)4月16日

(71)出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72)発明者 山口 龍一

大阪府門真市大字門真1006番地 松下電器
産業株式会社内

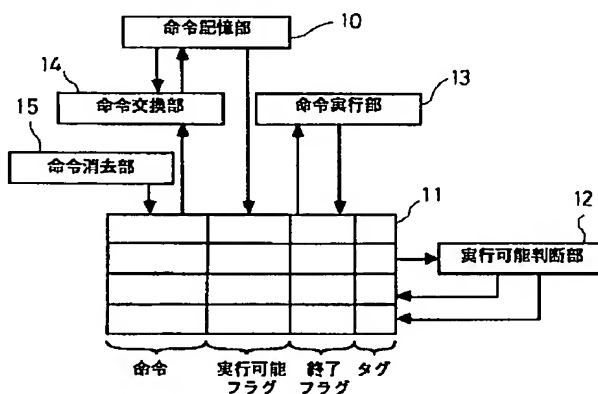
(74)代理人 弁理士 前田 弘 (外2名)

(54)【発明の名称】 情報処理装置

(57)【要約】

【目的】実行不可能である命令で実行制御部が一杯になる頻度を低下させ、命令を実行制御部へ円滑に供給して命令実行効率を向上させる。

【構成】 実行制御部11に読み込まれた命令は、実行可能判断部12により実行可能かどうか判断され、該命令が実行制御部11中の未実行の他の命令との依存関係により実行不能であれば、実行制御部11のタグがセットされる。実行制御部11中のタグがセットされた命令と実行順序が後の他の命令とが交換可能であれば、命令交換部14は、両者の格納順序を交換するように命令記憶部10への書き込みを実行する。この交換の結果、同じプログラムを再実行する場合、実行不能の命令に先立って、実行可能な命令が実行制御部11へ送られる。



【特許請求の範囲】

【請求項1】 命令記憶部に格納された複数の命令の実行順序の入れ換えが可能である情報処理装置であって、前記命令記憶部から読込んだ命令と、該命令の実行可能状態を示す実行可能フラグと、該命令の実行終了状態を示す終了フラグと、該命令と未実行の他の命令との間の実行の依存関係の存在を示すタグとを保持するための実行制御部と、前記実行制御部に保持された命令を読み込んで該命令が実行可能であるかどうかを判断し、該命令が実行可能であれば前記実行可能フラグをセットし、かつ該命令が前記実行制御部中の未実行の他の命令と依存関係にあるために実行不能であれば前記タグをセットするための実行可能判断部と、前記実行制御部の実行可能フラグにより実行可能であると示された命令を実行し、かつ該命令の実行終了後に該命令の前記終了フラグをセットするための命令実行部と、前記タグがセットされた命令が該命令より実行順序が後の他の命令と交換可能であれば該両命令の格納順序を交換するように前記命令記憶部への書き込みを実行するための命令交換部とを備えたことを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、命令記憶部から読み出された命令を命令実行部により実行する情報処理装置に関するものであり、とくに命令の実行順序の入れ換えが可能である情報処理装置に関するものである。

【0002】

【従来の技術】近年、情報処理装置の性能向上には著しいものがある。性能向上を図る手段として、同時に2個以上の命令を実行し、さらに命令間の実行依存関係を解消するため命令の実行順序を入れ換えることにより、命令処理速度を向上させる方法がある。

【0003】以下、図面を参照しながら、上記した従来の情報処理装置の一例について説明する。

【0004】図2は、実行順序を入れ換えて命令を実行する従来の情報処理装置のブロック図を示すものである。図2において、20は命令を記憶する命令記憶部である。21は命令記憶部20から読み込まれた命令と命令の実行可能状態を示す実行可能フラグと命令の実行終了状態を示す終了フラグとを保持する実行制御部である。22は実行制御部21に保持された命令を読み込んで実行可能であるかどうかを判断し、実行可能であれば実行可能フラグをセットする実行可能判断部である。23は実行制御部21の実行可能フラグにより実行可能であると示された命令を実行し、実行終了後に終了フラグをセットする命令実行部である。24は実行制御部21に保持された命令の中で終了フラグがセットされた命令

を実行制御部21から消去する命令消去部である。

【0005】図3は、実行される命令の例を示すプログラム図である。図3において、30、31、32、33はそれぞれ命令である。

【0006】以上のように構成された従来の情報処理装置について、以下その動作を説明する。

【0007】命令記憶部20に収められた命令を実行制御部21に読み込む。実行制御部21に読み込まれた命令は、実行可能判断部22により実行可能かどうか判断され、実行可能であれば実行制御部21の実行可能フラグがセットされる。命令実行部23は、実行制御部21中の実行可能フラグにより実行可能であると示された命令を実行し、この命令の実行終了後に実行制御部21の終了フラグをセットする。実行制御部21中の終了フラグがセットされた命令は、命令消去部24により実行制御部21から除去される。

【0008】例として、図3に示されたプログラムを実行する場合を取りあげる。第1～第4の命令30、31、32、33が命令記憶部20に収められているものとする。このうち3つの命令30、31、33はすぐに実行可能であるが、第3の命令32（102番地）は第2の命令31（101番地）が終了しないと実行できないものである。さらに、実行制御部21には合計3個の命令を保持できるものとする。

【0009】実行制御部21は命令記憶部20より第1～第3の命令30、31、32を読み込む。読み込まれた3つの命令30、31、32は実行可能判断部22により実行可能かどうか判断される。この例では、第3の命令32は第2の命令31が終了しないと実行できない。そこで、第1の命令30と第2の命令31の実行可能フラグのみがセットされる。実行可能フラグがセットされた2つの命令30、31は、命令実行部23に送られて実行される。実行が終了すれば実行制御部21の終了フラグがセットされ、命令消去部24により実行制御部21から実行終了した命令が除去される。

【0010】ここで第1の命令30だけが先に実行終了した場合、該第1の命令30が実行制御部21から消去され、第4の命令33が実行制御部21に送られる。この際、実行制御部21に収められた第2～第4の命令31、32、33の中で、実行可能判断部22により第4の命令33の実行可能フラグがセットされる。そして、第4の命令33が命令実行部23に送られ、第2の命令31と第4の命令33が実行中となる。以上のように、第3の命令32と第4の命令33の実行順序が入れ換えられて実行される。

【0011】

【発明が解決しようとする課題】しかしながら上記のような構成では、実行制御部に収める命令数には制限があり、実行できなかった命令で実行制御部が一杯になった場合には、実行可能な命令を命令記憶部から実行制御部

へ送ることができないという問題点がある。

【0012】本発明は上記問題点に鑑み、実行制御部が実行できなかった命令で一杯になる頻度を低下させ、命令を実行制御部へ円滑に供給できる情報処理装置を提供するものである。

【0013】

【課題を解決するための手段】上記問題点を解決するために本発明は、命令記憶部に格納された複数の命令の実行順序の入れ換えが可能である情報処理装置において、命令間に実行順序の依存関係がある場合には命令記憶部

10

中の命令の格納順序を変更することとしたものである。

【0014】具体的には、本発明は、次のような実行制御部と実行可能判断部と命令実行部と命令交換部とを備えた構成を採用したものである。すなわち、実行制御部は、命令記憶部から読込んだ命令と、該命令の実行可能状態を示す実行可能フラグと、該命令の実行終了状態を示す終了フラグと、該命令と未実行の他の命令との間の実行の依存関係の存在を示すタグとを保持するものである。実行可能判断部は、実行制御部に保持された命令を読み込んで該命令が実行可能であるかどうかを判断し、該命令が実行可能であれば実行可能フラグをセットし、かつ該命令が実行制御部中の未実行の他の命令と依存関係にあるために実行不能であればタグをセットするものである。命令実行部は、実行制御部の実行可能フラグにより実行可能であると示された命令を実行し、かつ該命令の実行終了後に該命令の終了フラグをセットするものである。また、命令交換部は、タグがセットされた命令が該命令より実行順序が後の他の命令と交換可能であれば該両命令の格納順序を交換するように命令記憶部への書き込みを実行するものである。

20

30

【0015】

【作用】本発明によれば、命令間の依存関係があるために実行できない命令を実行順序が後の他の命令と交換し、前者が実行制御部に送られるのを遅らせて、実行可能である命令を先に実行制御部に送ることにより、実行制御部が実行できなかった命令で一杯になる頻度を低下させ、命令を実行制御部へ円滑に供給することができる。

【0016】

【実施例】以下、本発明の一実施例の情報処理装置について、図面を参照しながら説明する。

【0017】図1は、本発明の実施例における情報処理装置のブロック図を示すものである。図1において、10は命令を記憶する命令記憶部である。11は命令記憶部10から読み込まれた命令と命令の実行可能状態を示す実行可能フラグと命令の実行終了状態を示す終了フラグと命令間の実行依存関係の存在を示すタグとを保持する実行制御部である。12は実行制御部11に保持された命令を読み込んで実行可能であるかどうかを判断し、実行可能であれば実行可能フラグをセットし、実行制御

50

部11中の未実行の命令と依存関係にあるために実行不能であればタグをセットする実行可能判断部である。13は実行制御部11の実行可能フラグにより実行可能であると示された命令を実行し、実行終了後に終了フラグをセットする命令実行部である。14は実行制御部11中のタグがセットされた命令と実行順序が後の命令とが交換可能であれば、両者の格納順序を交換するように命令記憶部10への書き込みを実行する命令交換部である。15は実行制御部11に保持された命令の中で終了フラグがセットされた命令を実行制御部11から消去する命令消去部である。

【0018】以上のように構成された本実施例の情報処理装置について、以下その動作を説明する。

【0019】命令記憶部10に収められた命令を実行制御部11に読み込む。実行制御部11に読み込まれた命令は、実行可能判断部12により実行可能かどうか判断され、実行可能であれば実行制御部11の実行可能フラグがセットされる。実行制御部11に読み込まれた命令が該実行制御部11中で未実行の他の命令と依存関係にあるために実行不能であれば、実行制御部11のタグをセットする。命令実行部13は、実行制御部11中の実行可能フラグにより実行可能であると示された命令を実行し、この命令の実行終了後に実行制御部11の終了フラグをセットする。実行制御部11中のタグがセットされた命令と実行順序が後の他の命令とが交換可能であれば、両者の格納順序を交換するように命令記憶部10への書き込みを実行する。その後、実行制御部11中の終了フラグがセットされた命令は、命令消去部15により実行制御部11から除去される。

【0020】例として、図3に示された前記のプログラムを実行する場合を取りあげる。また、前記従来例の場合と同様に、実行制御部11には合計3個の命令を保持できるものとする。

【0021】実行制御部11は命令記憶部10より第1～第3の命令30、31、32を読み込む。読み込まれた3つの命令30、31、32は実行可能判断部12により実行可能かどうか判断される。この例では、第3の命令32は第2の命令31が終了しないと実行できない。そこで、第1の命令30と第2の命令31の実行可能フラグのみがセットされる。また、第3の命令32は第2の命令31と依存関係にあるために実行不能であるので、第3の命令32のタグがセットされる。実行可能フラグがセットされた2つの命令30、31は、命令実行部13に送られて実行される。実行が終了すれば実行制御部11の終了フラグがセットされ、命令消去部15により実行制御部11から実行終了した命令が除去される。

【0022】ここで第1の命令30だけが先に実行終了した場合、該第1の命令30が実行制御部11から消去され、第4の命令33が実行制御部11に送られる。こ

の際、実行制御部11に収められた第2～第4の命令31、32、33の中で、実行可能判断部12により第4の命令33の実行可能フラグがセットされる。そして、第4の命令33が命令実行部13に送られ、第2の命令31と第4の命令33が実行中となる。ここで、実行制御部11中のタグがセットされた第3の命令32は実行順序が後の第4の命令33と交換可能であるので、命令交換部14は、これら両命令の格納順序を入れ換えるように命令記憶部10の内容を操作する。この結果、図4に示すように命令記憶部10には、第1の命令30と第2の命令31に続いて第4の命令33が102番地に保存され、該第4の命令33の次の103番地に第3の命令32が保存されることとなる。

【0023】従って、同じプログラムを再度実行する場合には、第1の命令30と第2の命令31と第4の命令33とが実行制御部11に読み込まれ、これら全ての命令の実行を命令実行部13により直ちに開始することができる。つまり、第2の命令31が実行終了するまで実行できない第3の命令32は実行制御部11へ送られるのが遅られ、実行制御部11が実行できなかった命令で一杯になる頻度を低下させることができる。

【0024】以上のように本実施例によれば、命令間の依存関係があるために実行できない命令を実行順序が後の命令と交換し、実行できない命令が実行制御部に送られるのを遅らせて、実行可能である命令を先に実行制御部に送ることにより、実行制御部が実行できなかった命令で一杯になる頻度を低下させ、命令を実行制御部へ円滑に供給することができる。

* 滑に供給することができる。

【0025】

【発明の効果】以上のように本発明によれば、命令間の依存関係があるために実行できない命令と実行順序が後の他の命令との格納順序を交換する構成を採用したので、前者が実行制御部へ送られるのを遅らせて、実行可能である命令を先に実行制御部へ送ることにより、実行制御部が実行できなかった命令で一杯になる頻度を低下させ、命令を実行制御部へ円滑に供給することができる。

【図面の簡単な説明】

【図1】本発明の実施例に係る情報処理装置のブロック図である。

【図2】従来の情報処理装置のブロック図である。

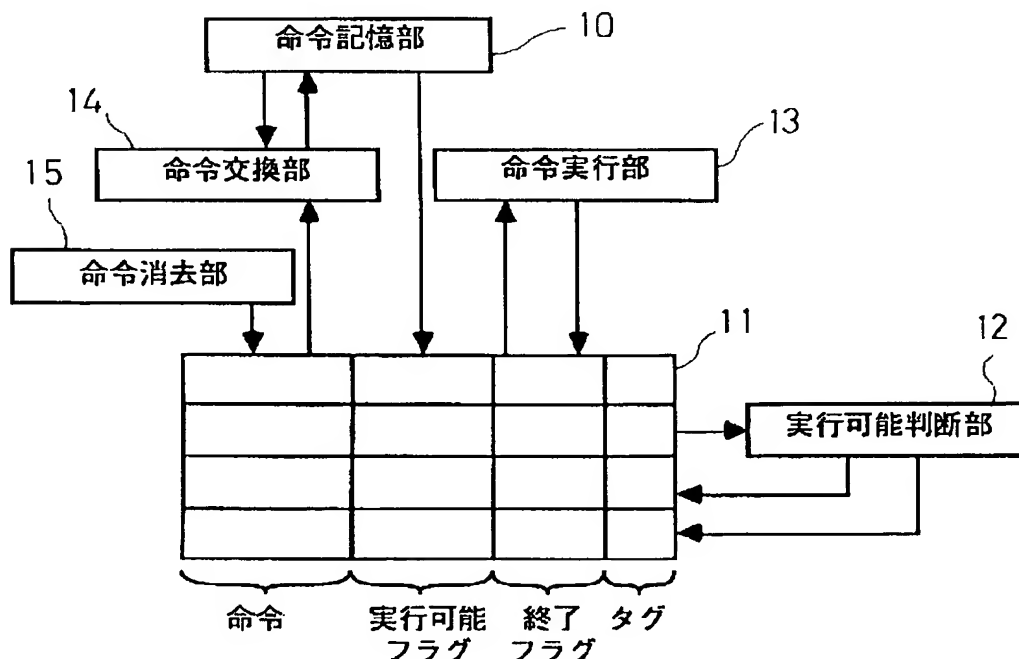
【図3】実行される命令の例を示すプログラム図である。

【図4】図3中の命令の格納順序が図1中の命令交換部により一部変更された結果を示すプログラム図である。

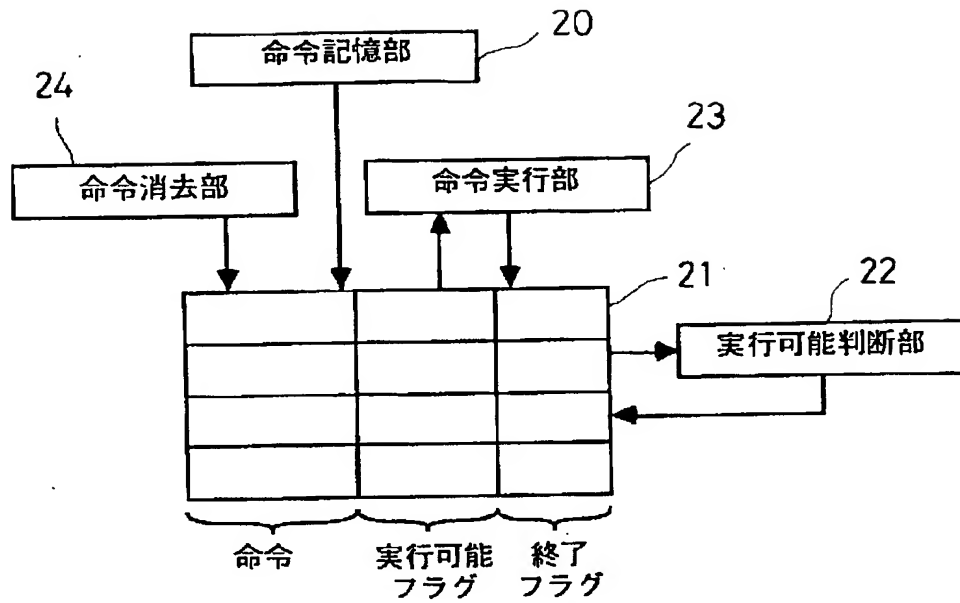
【符号の説明】

10、20 命令記憶部
11、21 実行制御部
12、22 実行可能判断部
13、23 命令実行部
14 命令交換部
15、24 命令消去部
30～33 命令

【図1】



【図2】



【図3】

100番地: ADD A, B — 30
 101番地: LOAD address_c, C — 31
 102番地: ADD C, D — 32
 103番地: ADD E, F — 33

【図4】

100番地: ADD A, B — 30
 101番地: LOAD address_c, C — 31
 102番地: ADD E, F — 33
 103番地: ADD C, D — 32

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] An information processor comprising:

A command which is an information processor which can replace execution sequence of two or more commands stored in an instruction storage part, and was read from said instruction storage part.

An executable flag which shows ready condition of this command.

An ending flag which shows execution exit status of this command.

An execution control part for holding a tag in which existence of a dependency of execution between this command and other commands of un-performing is shown, Read a command held at said execution control part, and it is judged whether this command can be executed, A judgment part for setting said tag, if impracticable, since said executable flag is set and other commands of un-performing in said execution control part and a dependency have this command, if this command can be executed which can be performed, An instruction-execution part for executing a command shown [that it can perform with an executable flag of said execution control part, and], and setting said ending flag of this command after an end of execution of this command, A command exchange section for performing writing to said instruction storage part so that storing order of these commands of both may be exchanged, if a command for which said tag was set is as more nearly exchangeable as a command of everything [execution sequence] but the back than this command.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention relates to the information processor which executes the command read from the instruction storage part by an instruction-execution part.

It is related with the information processor in which especially exchange of the execution sequence of a command is possible.

[0002]

[Description of the Prior Art]In recent years, some improved efficiency of an information processor is remarkable. In order to execute two or more commands simultaneously and to cancel the execution dependency between commands further as a means to plan improved efficiency, there is a method of raising instruction-processing speed by replacing the execution sequence of a command.

[0003]Hereafter, an example of the above-mentioned conventional information processor is explained, referring to Drawings.

[0004]Drawing 2 shows the block diagram of the conventional information processor which replaces execution sequence and executes a command. In drawing 2, 20 is an instruction storage part which memorizes a command. 21 is an execution control part holding the ending flag which shows the execution exit status of an executable flag and a command which shows the ready condition of the command and command which were read from the instruction storage part 20. It judges whether 22 can read the command held at the execution control part 21, and can be performed, and if execution is possible, it is a judgment part which sets an executable flag and which can be performed. 23 is an instruction-execution part which executes the command shown [that it can perform with the executable flag of the execution control part 21, and], and sets an ending flag after the end of execution. 24 is a command erasing part which eliminates the command in which the ending flag was set in the command held at the execution control part 21 from the execution control part 21.

[0005]Drawing 3 is a program figure showing the example of the command executed. In drawing 3, 30, 31, 32, and 33 are commands, respectively.

[0006]About the conventional information processor constituted as mentioned above, the operation is explained below.

[0007]The command stored in the instruction storage part 20 is read into the execution control part 21. It is judged whether the command read into the execution control part 21 can be executed by the judgment part 22 which can be performed, and if execution is possible, the executable flag of the execution control part 21 will be set. The instruction-execution part 23 executes the command shown [that it can perform with the executable flag in the execution control part 21, and], and sets the ending flag of the execution control part 21 after the end of execution of this command. The command in which the ending flag in the execution control part 21 was set is removed from the execution control part 21 by the command erasing part 24.

[0008]The case where the program shown in drawing 3 is executed as an example is taken up. The 1st - the 4th command 30, 31, 32, and 33 shall be stored in the instruction storage part 20. Among these, although three commands 30, 31, and 33 can be executed immediately, the 3rd command 32 (102nd street) cannot be executed unless the 2nd command 31 (101st street) is completed. A total of three commands shall be held in the execution control part 21.

[0009]The execution control part 21 reads the 1st - the 3rd command 30, 31, and 32 from the instruction storage part 20. It is judged whether three read commands 30, 31, and 32 can be executed by the judgment part 22 which can be performed. In this example, the 3rd command 32 cannot be executed, unless the 2nd command 31 is completed. Then, only the executable flag of the 1st command 30 and the 2nd command 31 is set. Two commands 30 and 31 in which the executable flag was set are sent to the instruction-execution part 23, and are executed. If

execution is completed, the ending flag of the execution control part 21 will be set, and the command which carried out the end of execution from the execution control part 21 by the command erasing part 24 is removed.

[0010] When only the 1st command 30 carries out the end of execution previously here, this 1st command 30 is eliminated from the execution control part 21, and the 4th command 33 is sent to the execution control part 21. Under the present circumstances, the executable flag of the 4th command 33 is set by the judgment part 22 which can be performed in the 2nd stored in the execution control part 21 - the 4th command 31, 32, and 33. And the 4th command 33 is sent to the instruction-execution part 23, and the 2nd the command 31 and the command 33 of the 4th become under execution. As mentioned above, execution sequence of the 3rd command 32 and the 4th command 33 is replaced and performed.

[0011]

[Problem(s) to be Solved by the Invention] However, in the above composition, when the number of commands stored in an execution control part has restriction and an execution control part fills with the command which was not able to be executed, there is a problem that a executable command cannot be sent to an execution control part from an instruction storage part.

[0012] This invention reduces the frequency which fills with the command which was not able to perform an execution control part in view of the above-mentioned problem, and provides the information processor which can supply a command to an execution control part smoothly.

[0013]

[Means for Solving the Problem] In order to solve the above-mentioned problem, when a dependency of execution sequence is between commands, suppose this invention that storing order of a command in an instruction storage part is changed in an information processor which can replace execution sequence of two or more commands stored in an instruction storage part. [0014] Specifically, composition provided with the following execution control parts, a judgment part which can be performed, an instruction-execution part, and a command exchange section is used for this invention. That is, an execution control part holds a tag in which existence of a dependency of execution between a command read from an instruction storage part, an executable flag which shows ready condition of this command, an ending flag which shows execution exit status of this command, and this command and other commands of un-performing is shown. It is judged whether the judgment part which can be performed can read a command held at an execution control part, and can execute this command, A tag will be set if impracticable, since an executable flag is set and other commands of un-performing in an execution control part and a dependency have this command, if this command can be executed. An instruction-execution part executes a command shown [that it can perform with an executable flag of an execution control part, and], and sets an ending flag of this command after an end of execution of this command. If a command for which a tag was set is as more nearly exchangeable as a command of everything [execution sequence] but the back than this command, a command exchange section will perform writing to an instruction storage part so that storing order of these commands of both may be exchanged.

[0015]

[Function] In this invention, execution sequence exchanges for other next commands the command which cannot be executed since there is a dependency between commands, it delays that the former is sent to an execution control part, and a executable command is previously sent to an execution control part.

Therefore, the frequency which fills with the command which was not able to perform an execution control part can be reduced, and a command can be smoothly supplied to an execution control part.

[0016]

[Example] Hereafter, the information processor of one working example of this invention is explained, referring to Drawings.

[0017] Drawing 1 shows the block diagram of the information processor in working example of this invention. In drawing 1, 10 is an instruction storage part which memorizes a command. 11 is an execution control part holding the tag in which existence of the ending flag which shows the execution exit status of an executable flag and a command which shows the ready condition of the command and command which were read from the instruction storage part 10, and the execution dependency between commands is shown. Since it judges whether 12 can read the command held at the execution control part 11, and can be performed, an executable flag will be set if execution is possible, and it is in a command of un-performing in the execution control part 11, and a dependency, if impracticable, it is a judgment part which sets a tag and which can be performed. 13 is an instruction-execution part which executes the command shown [that it can perform with the executable flag of the execution control part 11, and], and sets an ending flag after the end of execution. 14 is a command exchange section which performs the writing to the instruction storage part 10 so that both storing order may be exchanged, if a next command has exchangeable command and execution sequence with which the tag in the execution control part 11 was set. 15 is a command erasing part which eliminates the command in which the ending flag was set in the command held at the execution control part 11 from the execution control part 11.

[0018] About the information processor of this example constituted as mentioned above, the operation is explained below.

[0019] The command stored in the instruction storage part 10 is read into the execution control part 11. It is judged whether the command read into the execution control part 11 can be executed by the judgment part 12 which can be performed, and if execution is possible, the executable flag of the execution control part 11 will be set. Since other commands of un-performing and a dependency have the command read into the execution control part 11 in this execution control part 11, if impracticable, the tag of the execution control part 11 will be set. The instruction-execution part 13 executes the command shown [that it can perform with the executable flag in the execution control part 11, and], and sets the ending flag of the execution control part 11 after the end of execution of this command. If the command in which the tag in the execution control part 11 was set, and the next command of others [execution sequence] are exchangeable, the writing to the instruction storage part 10 will be performed so that both storing order may be exchanged. Then, the command in which the ending flag in the execution control part 11 was set is removed from the execution control part 11 by the command erasing part 15.

[0020] The case where the aforementioned program shown in drawing 3 is executed as an example is taken up. In the execution control part 11, a total of three commands shall be held like the case of said conventional example.

[0021] The execution control part 11 reads the 1st - the 3rd command 30, 31, and 32 from the instruction storage part 10. It is judged whether three read commands 30, 31, and 32 can be executed by the judgment part 12 which can be performed. In this example, the 3rd command 32 cannot be executed, unless the 2nd command 31 is completed. Then, only the executable flag of

the 1st command 30 and the 2nd command 31 is set. Since the 2nd the command 31 and dependency have the 3rd command 32 and it is impracticable, the tag of the 3rd command 32 is set. Two commands 30 and 31 in which the executable flag was set are sent to the instruction-execution part 13, and are executed. If execution is completed, the ending flag of the execution control part 11 will be set, and the command which carried out the end of execution from the execution control part 11 by the command erasing part 15 is removed.

[0022]When only the 1st command 30 carries out the end of execution previously here, this 1st command 30 is eliminated from the execution control part 11, and the 4th command 33 is sent to the execution control part 11. Under the present circumstances, the executable flag of the 4th command 33 is set by the judgment part 12 which can be performed in the 2nd stored in the execution control part 11 - the 4th command 31, 32, and 33. And the 4th command 33 is sent to the instruction-execution part 13, and the 2nd the command 31 and the command 33 of the 4th become under execution. Here, in the 3rd command 32 in which the tag in the execution control part 11 was set, since execution sequence is as exchangeable as the 4th next command 33, the command exchange section 14 operates the contents of the instruction storage part 10 so that the storing order of these both commands may be replaced. As a result, as shown in drawing 4, the 4th command 33 will be saved at the 102nd street at the instruction storage part 10 following the 1st the command 30 and the command 31 of the 2nd, and the 3rd command 32 will be saved at the 103rd street next to this 4th command 33.

[0023]Therefore, when executing the same program again, the 1st the command 30, the 2nd command 31, and the command 33 of the 4th are read into the execution control part 11, and execution of all a command of these can be promptly started by the instruction-execution part 13. That is, it is delayed that the 3rd command 32 that cannot be executed until the 2nd command 31 carries out the end of execution is sent to the execution control part 11, and it can reduce the frequency which fills with the command which was not able to perform the execution control part 11.

[0024]According to this example, execution sequence exchanges for a next command the command which cannot be executed since there is a dependency between commands as mentioned above, By delaying that the command which cannot be executed is sent to an execution control part, and sending a executable command to an execution control part previously, the frequency which fills with the command which was not able to perform an execution control part can be reduced, and a command can be smoothly supplied to an execution control part.

[0025]

[Effect of the Invention]By this invention, the composition which exchanges the storing order of the command which cannot be executed since there is a dependency between commands, and a command of everything [execution sequence] but the back was adopted as mentioned above. Therefore, by delaying that the former is sent to an execution control part, and sending a executable command to an execution control part previously, the frequency which fills with the command which was not able to perform an execution control part can be reduced, and a command can be smoothly supplied to an execution control part.

[Translation done.]